

### IN THE SPECIFICATION

Please amend the last paragraph on page 11 through the first paragraph on page 12 as follows:

Rule #4 can be used to logically negate a structured assembly language expression, *i.e.*, !SA\_Expr, to form a structured assembly language expression, by:

1. the branch label for all tuples in expression 1, except for the last tuple is complemented ~~complemented~~ (*i.e.*, next\_or becomes next\_and, next\_and become next\_or, otherwise, no change); and
2. the condition code for the last tuple of the expression is inverted.

Please amend the second paragraph on page 12 as follows:

For example, in order to assemble the negated version of the previous structured assembly language expression, that is;

```
!( ((arg1a <cc1> arg1b) && (arg2a <cc2> arg2b))  
  || ((arg3a <cc3> arg3b) || (arg4a <cc4> arg4b)) )
```